University of Dayton
eCommons

Computer Science Faculty Publications | Department of Computer Science

2009

Exploring Out-of-turn Interactions with Websites

Saverio Perugini
University of Dayton, sperugini1@udayton.edu

Naren Ramakrishnan
Virginia Polytechnic Institute and State University

Manuel A. Pérez-Quiñones
Virginia Polytechnic Institute and State University

Mary E. Pinney
Virginia Polytechnic Institute and State University

Mary Beth Rosson
Pennsylvania State University - Main Campus

Follow this and additional works at: https://ecommons.udayton.edu/cps_fac_pub

 Part of the Databases and Information Systems Commons, Graphics and Human Computer Interfaces Commons, Other Computer Sciences Commons, Systems Architecture Commons, and the Theory and Algorithms Commons

eCommons Citation

Perugini, Saverio; Ramakrishnan, Naren; Pérez-Quiñones, Manuel A.; Pinney, Mary E.; and Rosson, Mary Beth, "Exploring Out-of-turn Interactions with Websites" (2009). *Computer Science Faculty Publications*. 15.
https://ecommons.udayton.edu/cps_fac_pub/15

This Article is brought to you for free and open access by the Department of Computer Science at eCommons. It has been accepted for inclusion in Computer Science Faculty Publications by an authorized administrator of eCommons. For more information, please contact frice1@udayton.edu, mschlangen1@udayton.edu.

Exploring Out-of-turn Interactions with Websites

Saverio Perugini
Department of Computer Science
University of Dayton
Dayton, OH 45469-2160, USA
saverio@udayton.edu

Naren Ramakrishnan, Manuel A. Pérez-Quiñones
Department of Computer Science
Virginia Tech
Blacksburg, VA 24061, USA
{naren, perez}@vt.edu

Mary E. Pinney
Department of Industrial and Systems Engineering
Virginia Tech
Blacksburg, VA 24061, USA
mpinney@vt.edu

Mary Beth Rosson
School of Information Sciences and Technology
The Pennsylvania State University
University Park, PA 16802, USA
mrosson@ist.psu.edu

Abstract

Hierarchies are ubiquitous on the web, for structuring online catalogs and indexing multi-dimensional attributed datasets. They are a natural metaphor for information seeking if their levelwise structure mirrors the user's conception of the underlying domain. In other cases, they can be frustrating, especially if multiple drill-downs are necessary to arrive at information of interest. To support a broad range of users, site designers often expose multiple faceted classifications or provide within-page pruning mechanisms. We present a new technique, called *out-of-turn interaction*, that increases the richness of user interaction at hierarchical sites, without enumerating all possible completion paths in the site design. Using out-of-turn interaction, the user has the option to circumvent any navigation order imposed by the site and flexibly supply partial input that is otherwise relevant to the task. We conducted a user study to determine if and how users employ out-of-turn interaction, through a user interface we built called *Extempore*, for information-finding tasks. Extempore accepts out-of-turn input through voice or text and we employed it in a US congressional website for this study. Think-aloud protocols and questionnaires were utilized to understand users' rationale for choosing out-of-turn interaction. The results indicate that users are adept at discerning when out-of-turn interaction is necessary in a particular task, and actively interleaved it with browsing. However, users found cascading information across information-finding subtasks challenging. By empowering the user to supply unsolicited information while browsing, out-of-turn interaction bridges any mental mismatch between the user and the site. Our study not only improves our understanding of out-of-turn interaction, but also suggests further opportunities to enrich browsing experiences for users.

Introduction

Hierarchies are ubiquitous on the web, for structuring online catalogs and indexing multi-dimensional attributed datasets. They are a natural metaphor for information seeking if their levelwise structure mirrors the user's conception of the underlying domain, e.g., in a faceted manner. For instance, consider a hierarchical US Congressional Website, where the user progressively makes choices of politician attributes - *state* at the first level, *branch* at the second level, followed by levels for *party*, and *district/seat* - by browsing (see Fig. 1). At each level, the user browses by clicking on a presented hyperlink to communicate *partial information* about a desired congressperson, and the order in which the partial information arrives is pre-determined by the site designer. We say that browsing involves *in-turn* interaction. By in-turn, we mean that the user's input is responsive to the current solicitation of the site.

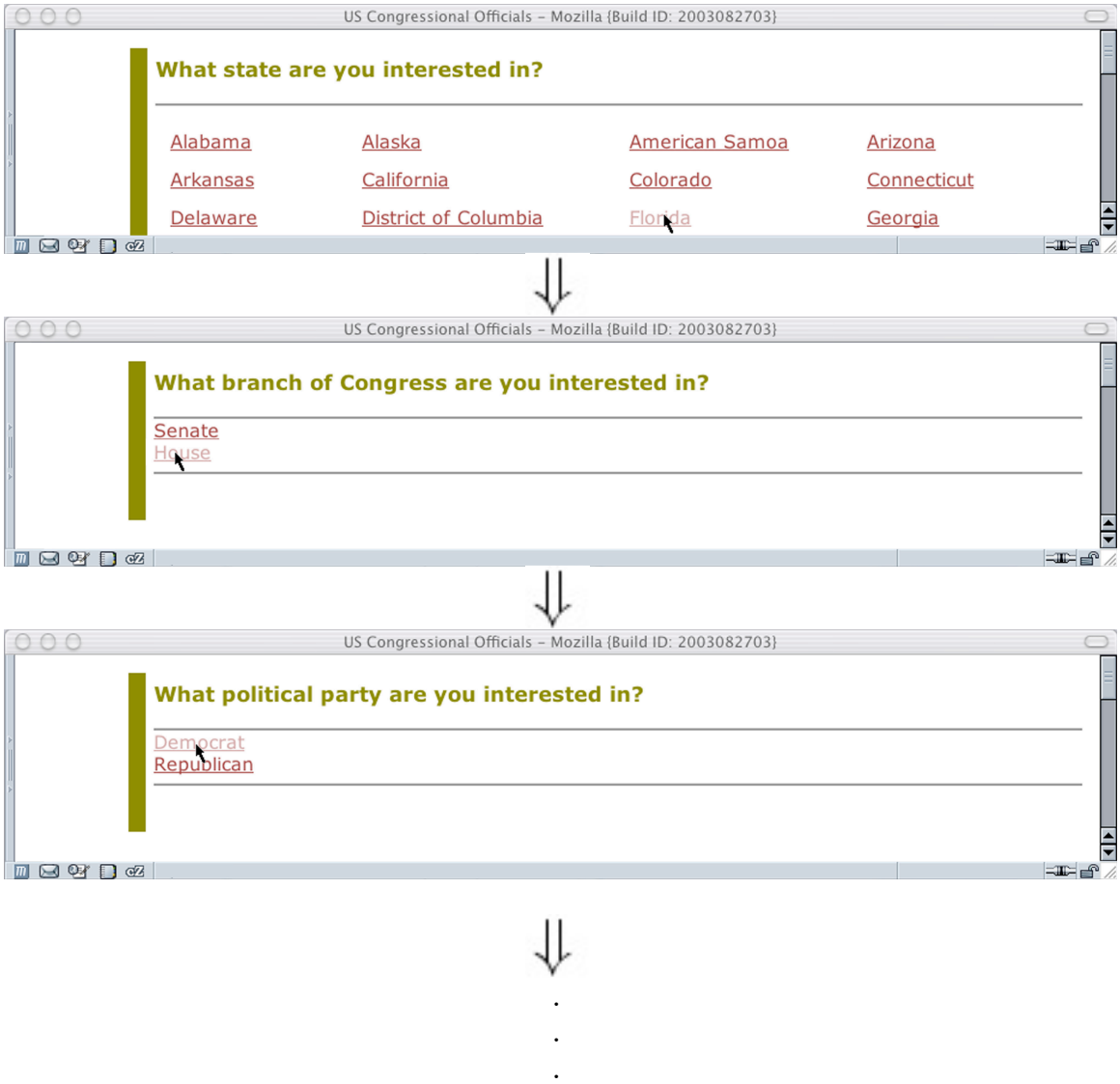


Figure 1: In-turn interaction with a Website.

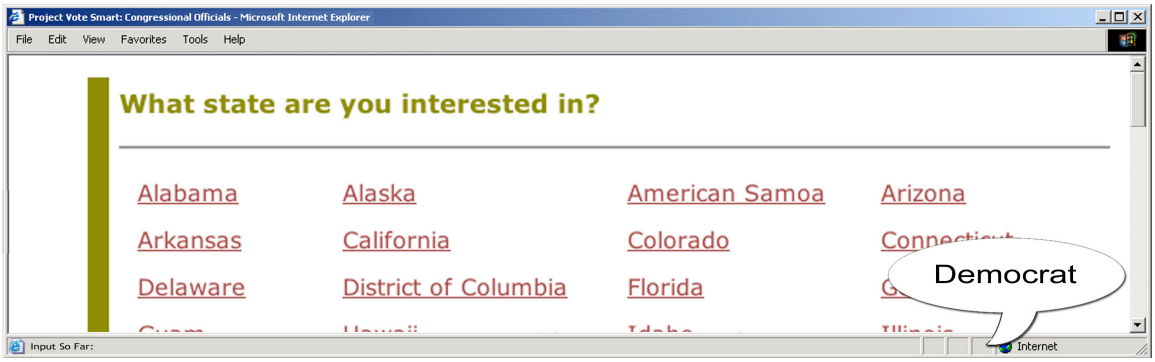
To increase the richness of user interaction, websites typically provide *multiple* faceted browsing classifications, especially when the facets can be communicated independent of each other. A common example can be seen in epicurious.com, described in (Hearst 2000). Directly supporting all permutations of facets in the browsing structure in this manner results in a cumbersome site design, with a mushrooming of choices at each step.

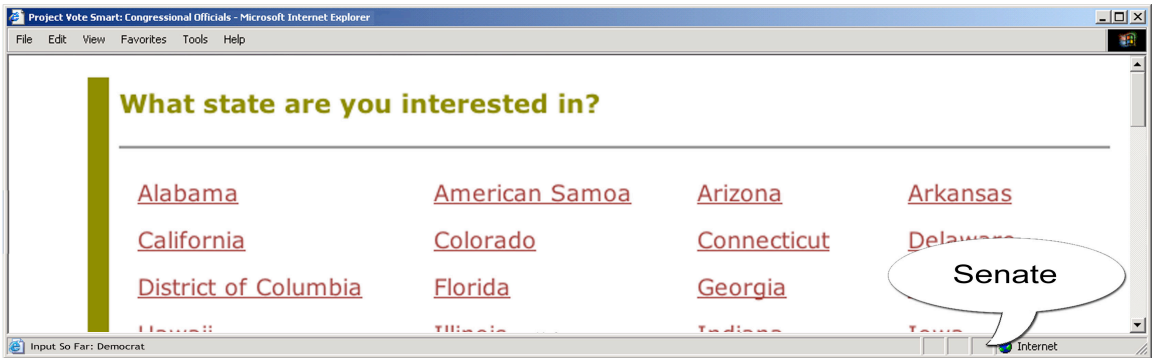
In this article, we introduce *out-of-turn* interaction -- a technique that increases the richness of user interaction at faceted sites, without enumerating all possible completion paths in the site design. Out-of-turn interaction is a technique that empowers the user to supply unsolicited information while browsing, and helps flexibly bridge any mental mismatch between the user and the site. We distinguish this technique from the traditional solutions for customizing information access, such as personalized search (Pitkow *et al.* 2002) and sites with multiple faceted browsing classifications (Hearst *et al.* 2002). We also present a study to explore the use of out-of-turn interaction in faceted Websites.

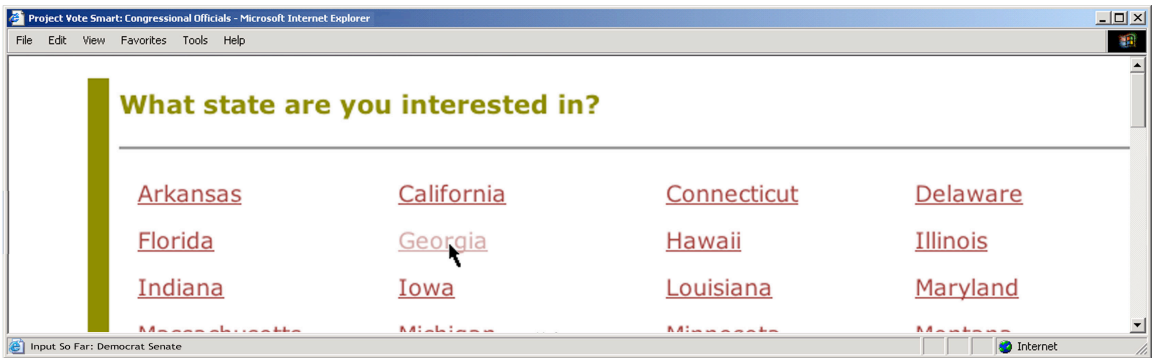
Out-of-turn interaction

Consider finding all the Democratic Senators in the site of Fig. 1. Achieving this task by browsing will require a painful series of drill-downs and roll-ups, in order to identify the states that have at least one Democratic Senator, and to aggregate the results. The site is requiring the user to specify partial information in terms of state, whereas the user has partial information about the congresspeople in terms of party and branch of Congress.

Out-of-turn (out-of-turn) interaction is our solution to support flexible communication of partial information not currently requested by the system. Hence such information is unsolicited but presumably relevant to the information-seeking task. out-of-turn interaction is thus unintrusive, optional, and can be introduced at multiple points in a browsing session, at the user's discretion. One possible means to support it is to allow the speaking of terms into the browser.




Project Vote Smart: Congressional Officials - Microsoft Internet Explorer
File Edit View Favorites Tools Help
What state are you interested in?
Alabama
American Samoa
Arizona
Arkansas
California
Colorado
Connecticut
District of Columbia
Florida
Georgia
Senate
Input So Far: Democrat
Internet


Project Vote Smart: Congressional Officials - Microsoft Internet Explorer
File Edit View Favorites Tools Help
What state are you interested in?
Arkansas
California
Connecticut
Delaware
Florida
Georgia
Hawaii
Illinois
Indiana
Iowa
Louisiana
Maryland
Input So Far: Democrat Senate
Internet

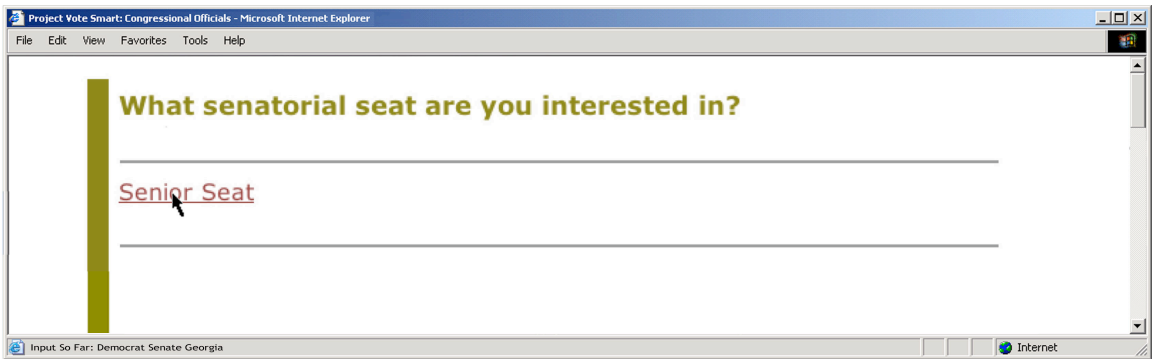

Project Vote Smart: Congressional Officials - Microsoft Internet Explorer
File Edit View Favorites Tools Help
What senatorial seat are you interested in?
Senior Seat
Input So Far: Democrat Senate Georgia
Internet



Figure 2: A Web session illustrating the use of out-of-turn interaction in a US congressional site. This progression of interactions shows how the (Democrat, Senate, Georgia, Senior) interaction sequence, which is indescribable by browsing, may be realized. In steps 1 and 2, `Democrat' and `Senate' are spoken out-of-turn, respectively, when the systems solicits for state. In step 3, the user clicks `Georgia' as the state (an in-turn input). The screen at step 4 shows that only the Senior Senator from Georgia is a Democrat, and leads the user to his homepage.

Fig. 2 showcases the use of out-of-turn interaction. At the top level of the site, the user is unable to make a choice of state, because s/he is looking for states that have Democratic Senators. S/he thus speaks `Democrat' out-of-turn, causing some states to be pruned out (e.g., Alaska). At the second step, the site again solicits state information because this aspect has not yet been communicated by the user. The user speaks `Senate' out-of-turn, causing further pruning (e.g., of American Samoa), and retaining only regions that have Democratic Senators. At this point, the goal has been achieved (the user notices 31 states satisfying the criteria), and s/he proceeds to browse through the remaining hyperlinks. Notice that these are contextually relevant to the partial information supplied thus far, so that when `Georgia' is clicked, there is only one choice of seat (Senior) implying that the other Senatorial seat is not occupied by a Democrat.

How is out-of-turn interaction different from using a search engine?
Search engines are characterized by specification of complete information, because the interaction is terminated by returning a flat list of results, which curbs the user-site dialog. out-of-turn interaction *continues* the dialog and situates future dialog choices (e.g., hyperlink options) in the context of previously supplied partial information.

How do users know what to say?

As Yankelovich (1996) points out, the typical problem in conversational engines is that `the functionality of applications is hidden, and the boundaries of what can and cannot be [said] are invisible.' Out-of-turn interaction inherits this problem but in a

less pronounced manner, since it merely allows a site's existing navigation structure to be realized in a *different order*. The only legal inputs are those that are already present in the browsing structure! Vocabulary for out-of-turn interaction is hence the same as the universal set of hyperlink labels. In the study presented here, the users were provided with meta-query facilities to help understand the yet-unspecified aspects of the dialog.

Why is out-of-turn interaction useful?

out-of-turn is useful because what the site is requesting from the user may actually be what the user is seeking in the first place! For example, in Fig. 2, the site is soliciting state but the user is looking for states with a certain property.

When is out-of-turn interaction relevant?

The benefits of out-of-turn interaction are immediately apparent when there are dependencies underlying the facets. For instance, not all states have Democrats, and so upon saying `Democrat' out-of-turn in Fig. 2, the user is immediately provided visual feedback by a pruning of state choices. Such pruning not only updates the dialog state but also implicitly reinforces the continuation of the dialog to the user. When there are a significant number of dependencies, and/or the site is non-faceted (e.g., the Open Directory Project at dmoz.org), such pruning is even more drastic and out-of-turn interaction effectively delivers a shortcut to a page deeper in the hierarchy. In this paper, we focus on faceted Websites.

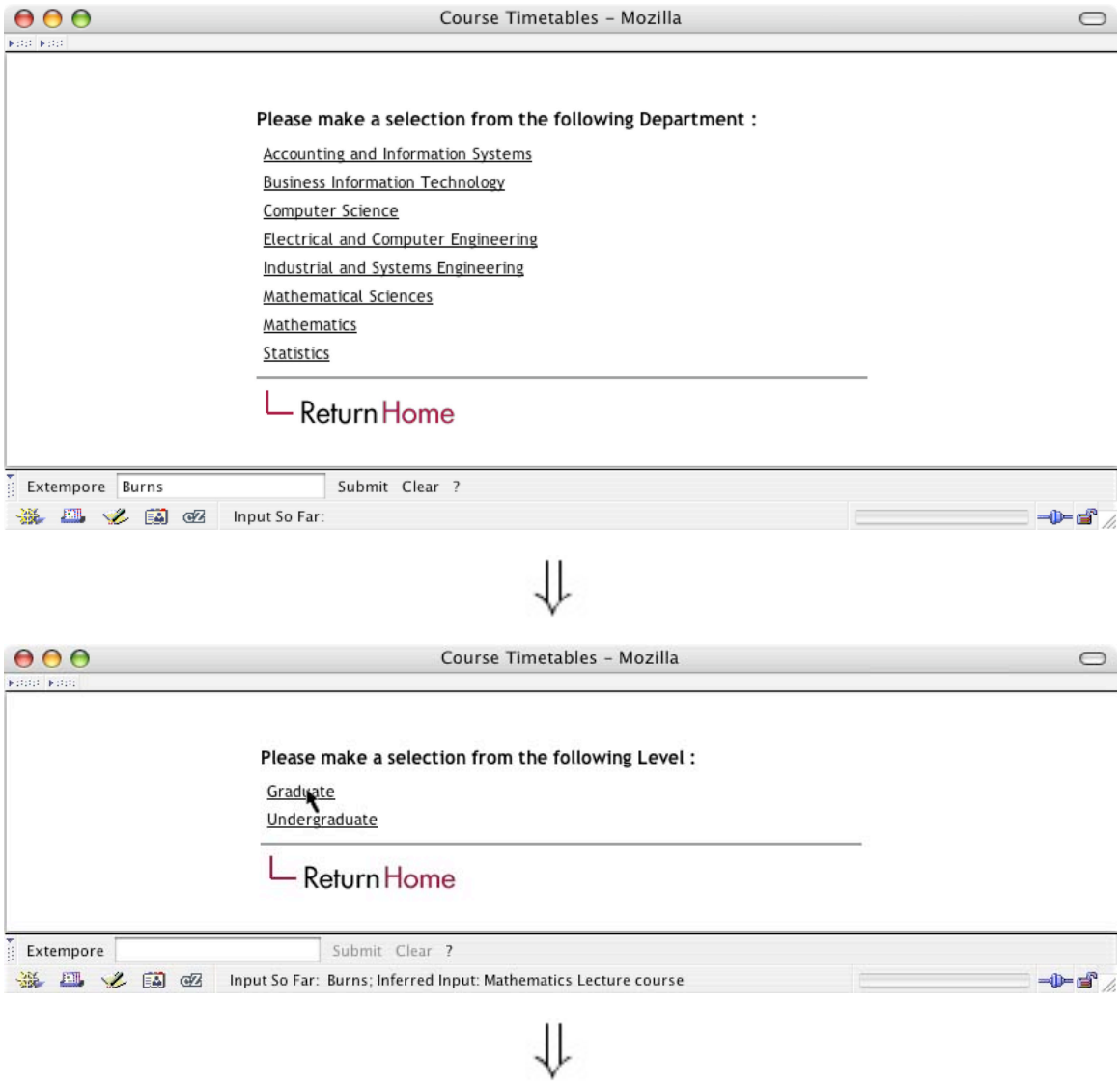
How can we study out-of-turn interactions?

out-of-turn interaction is a novel way to improve information access on the Web, and contrasts with ideas such as combining browsing and searching, e.g., ScentTrails (Olston and Chi 2003) and automated browsing, e.g., Letizia (Lieberman, Fry and Weitzman 2001). Before we compare out-of-turn interaction with these paradigms, we must first determine if users are effective at relating the nature of an information-seeking task with the affordances of out-of-turn interaction. This paper presents a study to explore this question. **The goal of this paper hence is to study usage patterns for out-of-turn interaction, not to evaluate the interfaces used to realize it, or to compare out-of-turn interaction with other interaction techniques.**

Extempore

We have built a user interface, called *Extempore*, that accepts out-of-turn input either through voice or text. The voice version was implemented using SALT 1.1 (Speech Application Language Tags 2002), a standard that augments HTML with tags for speech input/output, and SRGS (Speech Recognition Grammar Specification), for use with Internet Explorer 6.0. The text version is a toolbar embedded into the Mozilla FireFox Web browser (see Fig. 4) and was implemented using XUL (XML User Interface Language). At the time of our study, a SALT plugin was not available for Mozilla. Now the OpenSALT project at CMU makes available a SALT 1.0 compliant open-source browser based on Mozilla. Currently, XUL is not supported by IE. Due to these technological constraints, we did not support both

interfaces of Extempore in the same implementation. It is important to note that Extempore is embedded in the Web browser, and not the site's Webpages. It is also not a site-specific search tool that returns a flat list of results (akin to the Google toolbar). Further, while search engines index Webpages, Extempore relies on an internal representation of the Website and, when out-of-turn input is supplied, uses transformation techniques to stage the interaction, pruning the Website accordingly. The details of the underlying software transformations are beyond the scope of this work; see, e.g., Perugini and Ramakrishnan (2003) and Ricca and Tonella (2001) for ideas on transformation techniques. Extempore can be used for out-of-turn interaction in many Websites, given a representation of the site's structure, e.g., in XML.



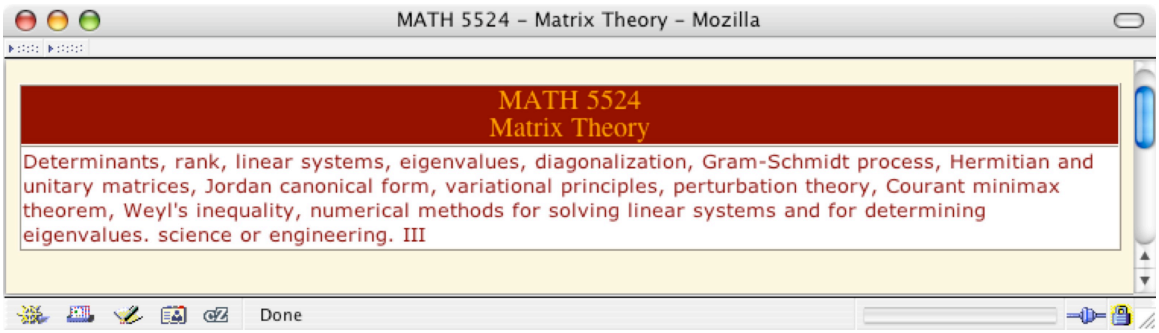


Figure 4: An out-of-turn dialog with an online course selection system. When prompted for a department, the user types in the name of an instructor (Burns) using the Extempore toolbar. Several dependencies are automatically triggered and used to infer that the user is interested in a lecture course offered by the Mathematics department. This effectively provides a shortcut to a choice of graduate versus undergraduate courses.

Exploratory study

Extempore was implemented for two sites: Project Vote Smart (PVS) (www.vote-smart.org) and an online university course selection system (see Fig. 4). Here we present the results of a user study using the former. At the time this study was conducted, PVS employed a hardwired browsing organization akin to that shown in Fig. 1; the site has since been restructured into a flat faceted classification. The study exposed users to out-of-turn interaction, to determine if they utilize it in information-finding tasks, and their rationale for doing so. The main component of the study entailed asking participants to perform eight specific information-finding tasks, and gathering rationale through think-aloud and retrospective protocols.

Participants

We collected data from 24 participants in the analysis; all were students with an average age of 21, and a majority were undergraduates in computer science. Some of the participants were recruited from a HCI course, and were compensated with extra-credit from the instructor. Average participant computer and Web familiarity and use was 4.75 or greater on a 5-point Likert scale. Average participant familiarity with voice recognition software was 1.46, and mean familiarity with the structure of the US Congress was 2.83; no user had visited the PVS Website before the experiment.

Tasks

The eight tasks were carefully formulated to generate a diverse set of interaction choices:

A. Find the Webpage of the Junior Senator from New York.
B. Find the Webpage of the Democratic Representative from District 17 of Florida.
C. Find the Webpage of the Republican Junior Senator from Oregon.
D. Find the Webpage of the Democratic member of the House in Rhode Island serving District 2.

E. Find the states which have at least one Democratic Senator.
F. Find the states which have twenty or more congressional districts.
G. Find the states which have at least one Republican member of the House.
H. Find the political party of the Senior Senator representing the only state which has congresspeople from the Independent party.

We refer to tasks A, B, C, and D as *non-oriented* tasks, in that they can be performed as easily by employing solely in-turn interaction (i.e., in this case, hyperlinks), solely out-of-turn interaction (Extempore), or using a mixture of both. Out-of-turn interaction does not appear to be worthwhile with respect to these tasks because the effort required to perform them with out-of-turn interaction is commensurate with that of in-turn interaction. Tasks E, F, G, and H are *out-of-turn-oriented*, because they are difficult to perform with only in-turn interaction. Formally, we say an information-seeking task is out-of-turn-oriented if the minimum number of browsing interactions required to successfully complete it exceeds the maximum depth of the targeted Website; otherwise it is non-oriented.

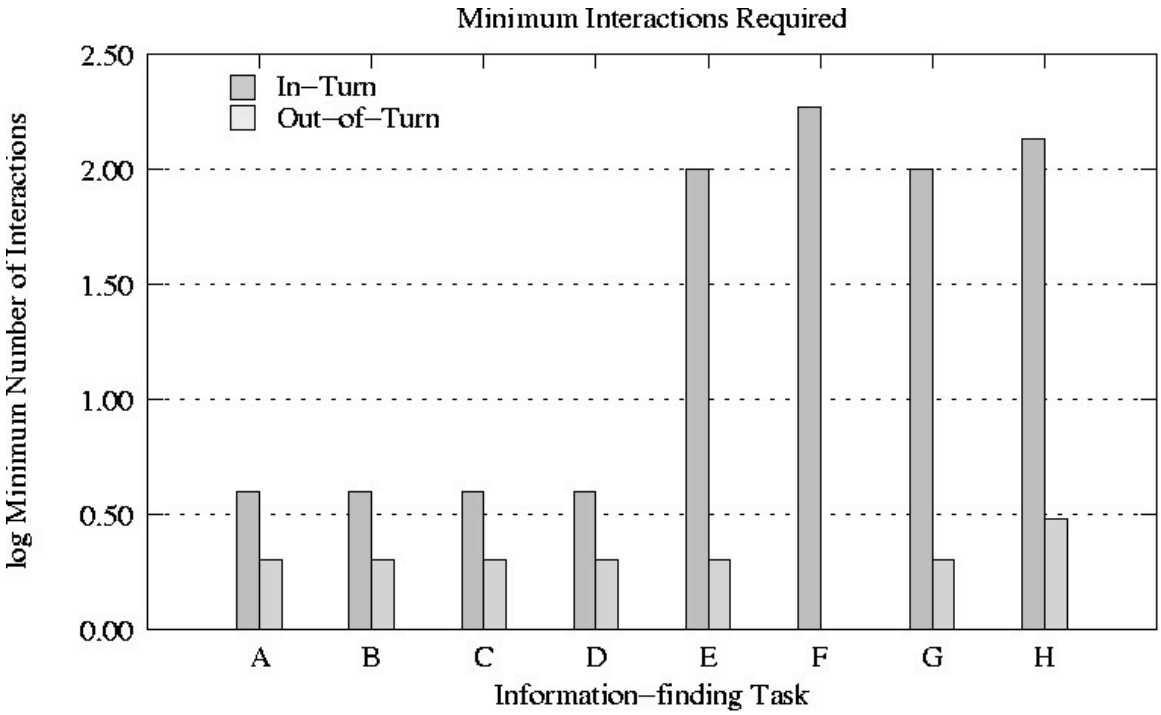


Figure 5: Minimum number of interactions (log_{10} scale) required to successfully satisfy each information-finding task using in-turn (dark) and out-of-turn (light) interaction. Note that Task F can be completed with just one out-of-turn interaction, so its entry in the graph shows zero.

The maximum depth of the PVS site is four and Fig. 5 illustrates the minimum number of interactions required per task. In calculating this minimum number, we assumed that the user can supply at most one input at each step (in-turn or out-of-turn), and discounted back button clicks (happens when employing only in-turn

interaction for an out-of-turn-oriented task). Notice also that some tasks, namely the non-oriented ones, cannot be performed by purely a sequence of out-of-turn interactions; a terminal in-turn input is often necessary and these are discounted as well. For instance, try solving task A using purely out-of-turn inputs.

Design

The study was designed as a within-subjects experiment. Task was the independent variable and the interaction observed (in-turn versus out-of-turn) was the dependent variable. Participants were given both the toolbar and voice interface of Extempore; and performed four tasks with each (two non-oriented and two out-of-turn-oriented). We designed the experiment with the provision for interfaces in two different modalities, to more naturally assess the use of out-of-turn interaction independent of a particular interface for it. Each participant performed the eight tasks in an order pre-determined by a Latin square to control for unmeasured factors. In addition, the specific interface to be used (toolbar or voice) for a (task, participant) pair was determined *a priori* by complete counterbalancing within each task category. Thus, for each task, half of the twenty-four participants were given the toolbar interface and half the voice interface. The participants were free to utilize any strategy to complete the information-finding tasks, given Extempore and the available hyperlinks, and they were given unlimited time to complete each task.

Configuring Extempore

A vocabulary for the PVS site was created by collecting all link labels, synonyms (e.g., `Representatives' for `House'), and alternate forms of common utterances (e.g., `Senate', `Senator', `Senators'). Both the toolbar and voice version of Extempore supported this vocabulary, with the toolbar supporting abbreviations (e.g., CA for California), in addition. To keep users abreast of partial information supplied thus far (either by browsing or through Extempore), we continually updated an `Input So Far:' label in the browser status bar (see Fig. 2). We have found this to be valuable as a feedback mechanism in out-of-turn interaction, especially when the site has processed the out-of-turn input, but no pruning is visible at the top level (e.g., say `House' at the top level of Fig. 2 or `lecture course' in Fig. 4 (top). We also included a provision for the user to enquire about what partial information is left unspecified at any step. Access to this feature is provided through a `What May I Say?' button (labeled with a `?' and seen on the toolbar in Fig. 4 (top and middle) or utterance.

The semantics of out-of-turn interaction in information hierarchies required some practical implementation decisions. For instance, when the user speaks `Junior seat,' the specification of `Senate' can be automatically inferred by functional dependency. Another form of such *utterance expansion* occurs in response to single-valued options. For instance, in Fig. 2, one can argue that the choice of seat at the last step is really unnecessary, as there is only one option left (Senior). When only one path remains among the available options, we vertically collapse them and directly present the leaf page. This feature was not illustrated in Fig. 2 for ease of presentation, but we implemented it in our study. Notice, however, that no

information is lost during such collapsing, since terminal pages in PVS identify all pertinent attributes of congresspeople.

Equipment, training, and procedures

Equipment

Participants performed the tasks on an Extempore-enabled Pentium III workstation, connected to a 17'' monitor set at 2560x1024 resolution in 34-bit true color, running Windows 2000. We captured each participant's on-screen actions as well as their audio using the Camtasia screen and audio capture software while they performed the information-finding tasks. The resulting capture was used to aid participant recollection during the retrospective verbal protocol as well as in subsequent analysis (e.g., think-aloud). The Audacity audio recording application was used during the retrospective portion of the experiment to capture participant explanations. Data from the pre-questionnaire (demographics, computer familiarity) and post-questionnaires (rationale) was recorded on paper. Finally at the end of the entire experiment we transcribed and collated the data gathered from all sources to construct a complete record of each participant session, including interaction sequences followed per task. Each participant session lasted approximately 90 minutes.

Training

Prior to revealing the information-seeking tasks, we gave users specific training on (i) the PVS Website, including levels of classification, and interacting with it through hyperlinks; (ii) interacting with PVS using Extempore (both toolbar and voice interfaces); and (iii) interleaving hyperlink clicks with commissions through Extempore. We did not use terms such as *in-turn* or *out-of-turn* during training or elsewhere in the study. This is to prevent biasing of participants toward any intended benefits of Extempore, and also to help them conceptualize its functionality on their own. In other words, we simply trained users on how to employ the available interfaces (hyperlink and Extempore) for information seeking. After some self-directed exploration, users were given a short test consisting of four practice tasks (two with the toolbar and two with voice).

Procedures

After the users completed the training tasks, we administered the actual test involving tasks A-H above, and employed both concurrent (think-aloud) and retrospective protocols to elucidate rationale. A structured interview, including a post-questionnaire, was conducted to gather additional feedback. The entire experiment generated (24 x 8 =) 192 (participant, task) interaction sequences.

These sequences were then analyzed for frequencies of usage of in-turn versus out-of-turn interaction. For purposes of this study, we defined an in-turn interaction as a hyperlink click or the communication of in-turn partial information to the Website through Extempore. Notice that just saying 'Connecticut' will not qualify as an out-of-turn interaction, if the same choice was currently available as a hyperlink. Similarly, we defined an out-of-turn interaction to be the submission of one aspect of unsolicited partial information to the site. Supplying more than one aspect of

partial information to the site out-of-turn (e.g., saying `Democratic Senators') corresponds to multiple out-of-turn interactions.

Notice that a user may supply in-turn and out-of-turn information to the Website simultaneously through Extempore. For instance, in the top-level page in Fig. 2, the user might say `House, Florida, District 17, Democrat,' all at the outset. Observe that a permutation of this utterance exists --- `Florida, House, Democrat, District 17' --- that, if conducted incrementally, could imply a purely in-turn interaction. Such an interaction is thus viewed as having four in-turn inputs. On the other hand, consider a user who says `New York, Democrat' at the outset. There is no permutation with respect to the PVS site that permits viewing this utterance as comprising of purely in-turn input, and hence, it is classified as one in-turn input (`New York'), followed by an out-of-turn input (`Democrat'). This policy of counting does not favor (and actually deprecates) out-of-turn interaction.

Some users, after completing a given task through out-of-turn interaction, verified part of their results through in-turn interactions. This was confirmed through their retrospective feedback, and such in-turn interactions were discounted in the analysis.

Results

Of the 192 recorded interaction sequences, 177 of them involved the successful completion of the task by the participant. We analyze these 177 sequences first, followed by the remaining 15 sequences (which were all generated in response to out-of-turn-oriented tasks).

General usage patterns

Results indicate a high frequency of usage for out-of-turn interaction. 94.4% of the 177 sequences contained at least one out-of-turn interaction. In addition, every participant used out-of-turn interaction for at least 70% of the tasks, with 16 people using it in all tasks. Conversely, every task was performed with out-of-turn interaction by at least 80% of the participants, with 4 tasks enjoying out-of-turn interaction by all participants. These results are encouraging because Extempore usage is optional and not prompted by any indicator on a Webpage. Participants successfully completed the given tasks irrespective of the presented interface (voice or toolbar).

Classifying interaction sequences

The 177 interaction sequences were classified into five categories denoted by: (i) I, (ii) O, (iii) IO, (iv) OI, and (v) M. The I and O categories denote sequences comprised of purely in-turn or out-of-turn inputs, respectively. In IO sequences all in-turn inputs precede out-of-turn inputs (analogously, for OI). For instance, the interaction shown in Fig. 1 would be classified under I, and that in Fig. 2 is in OI. M (*mixed*) sequences are those which do not fall in the above categories. We posit that this classification provides insight into users' information-seeking strategies, and can be related to the nature of the information-finding task.

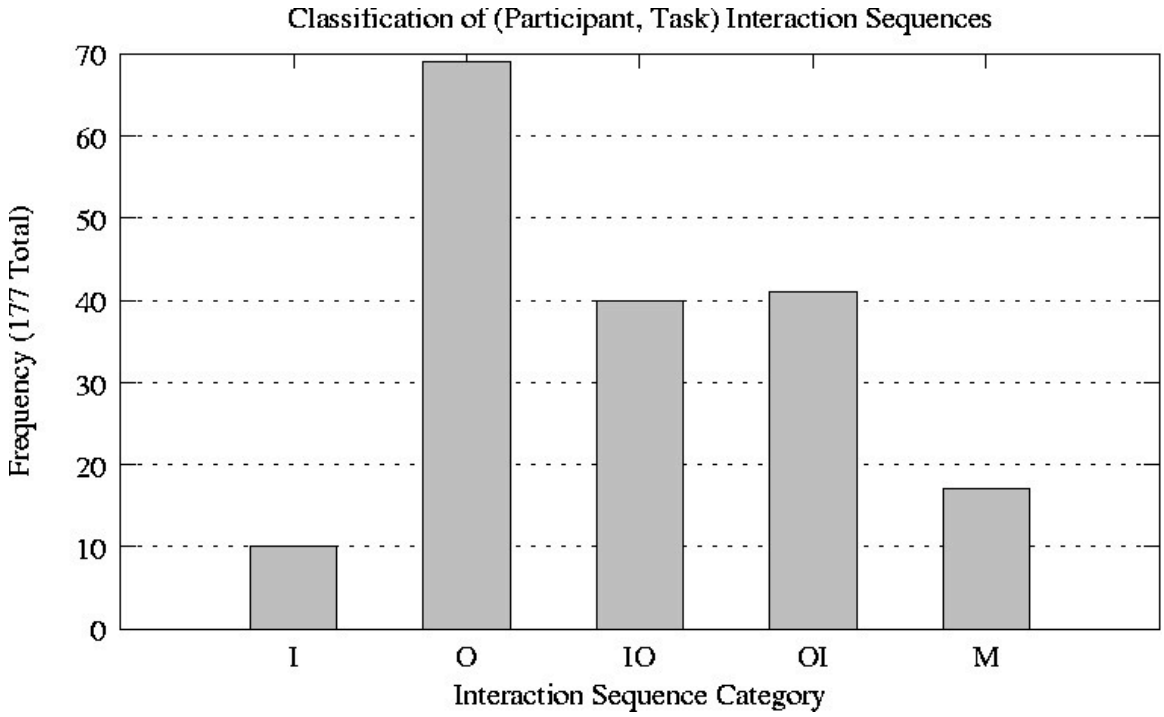


Figure 6: Classification of 177 (participant, task) interaction sequences.

	{I}	{O,IO,OI,M}	total
non-oriented	10	86	96
out-of-turn-oriented	0	81	81
total	10	167	177

Table 1: Breakdown of 177 interaction sequences in various categories. The total number of interaction sequences for out-of-turn oriented tasks is 15 less than that for non-oriented tasks; these were the sequences where the participant did not complete the task successfully.

Fig. 6 shows the distribution of the 177 sequences into the five classes, and Table 1 depicts a breakdown by both task orientation and classes. Notice that O, OI, IO, and M classes have been grouped in Table 1 to distinguish them from pure browsing interactions (I).

As Fig. 6 shows, 10 of the 177 sequences fall in the I class, i.e., these are *browsing* sequences. As Table 1 (lower left) shows, all of the 10 browsing sequences were generated in response to non-oriented tasks, revealing that a 100% (81/81) of the sequences for out-of-turn-oriented tasks involved out-of-turn interaction. Therefore,

1. users never attempted to achieve an out-of-turn-oriented task through browsing; or in other words,
2. users always employed out-of-turn interaction when presented with an out-of-turn-oriented task.

Our results show that users never succeeded at doing an out-of-turn-oriented task purely by browsing. This is notable because it confirms that users are adept at discerning when out-of-turn interaction is necessary.

Detailed analysis of interaction classes

Let us now study the interactions in classes O, OI, IO, and M. The 69 pure out-of-turn sequences (O) were observed only in out-of-turn-oriented tasks E, F, and G, and were used by all the 24 participants. This clustering of the O sequences on three tasks shows that, whenever participants completed these tasks, they did so in the shortest manner possible. Refer again to Fig. 5 for the sharp contrast in the length of the minimum out-of-turn sequence from the minimum in-turn sequence, for these tasks.

Classes IO, OI, and M contain the sequences exhibiting rich interaction strategies. Classes IO and OI were observed in near-equal numbers, and primarily in the non-oriented tasks (A, B, C, and D) with the exception of OI, which was also seen in task H. No particular clustering was observed with respect to participants. The 17 class M interactions exhibited only two types of patterns -- 14 with an OIO form, and 3 with an IOI form. Furthermore, like OI, these 17 mixed interactions also involved only the non-oriented tasks (A, B, C, D) and task H. It is interesting that we observed OIO and IOI sequences in a site with only four levels. Once again, no specific clustering was observed on participants.

Cascading information across subtasks

Recall that 15 interaction sequences led to incorrect answers; interestingly 12 of these 15 were generated in response to Task H. Notice that Task H is challenging, because it involves two subtasks and cascading information found in one into the other. The user is expected to first find the only state having Independent congressional officials (Vermont), and then find the political party of the Senior Senator from that state (Democrat). In other words, this task requires procedural, not just declarative, knowledge, a distinction motivated in the Strategy Hubs project (Bhavnani *et al.* 2003).

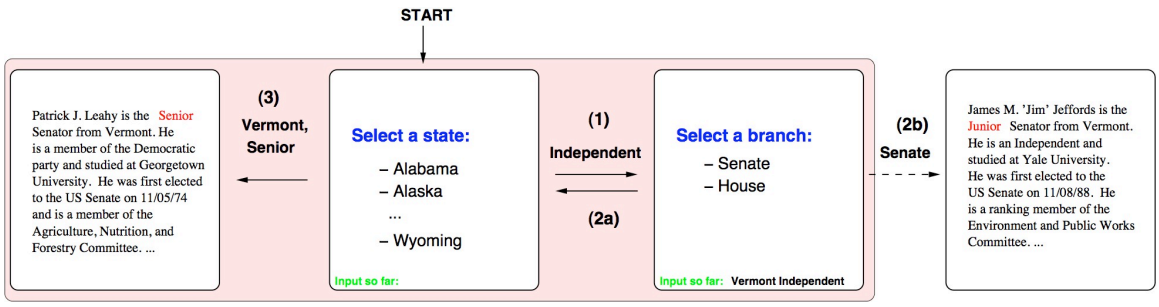


Figure 7: Task H: the user is expected to first find `Vermont' in one Interaction (1) and go back (2a) to supply it as input in another interaction (3) to find the party of the Senior Senator from that state. The interaction labeled 2b is unnecessary and irrelevant for this task.

Most participants were adept at finding that Vermont was the desired state (e.g., by saying `Independent' at the outset), but did not realize that the task cannot be completed by *continuing* that interaction. As Fig. 7 shows, clicking on the only available state link (`Vermont') now presents a choice of House versus Senate. Clicking on Senate takes the user to the Webpage of Jim Jeffords, who is the Junior Senator from Vermont, not the Senior Senator!

Some users immediately realized the problem, as identified in their retrospective interviews, e.g.:

> ``This question was tricky. Cause it was, I was like wait, if he's Independent then his party is Independent ... at first [I thought] it was the Senior Senator who was Independent ... and I got this guy's Webpage, and then I saw that he was the Junior ... So then I eventually went back to Vermont and got the [Senior] guy.''

Only 12 (50%) of the participants successfully completed this task. This result demonstrates that cascading information across subtasks is challenging. It was clear that all users wanted to continue the interaction, but some failed to realize that out-of-turn interaction as presented here is merely a pruning operator, and not constructive. Investigating the incorporation of constructive operators such as rollup/expansion is thus a worthwhile direction of future research.

Rationale and qualitative observations

Studying users' rationale revealed their reasons for interacting out-of-turn:

> ``I can jump through all the levels ....''
>
> ``Initially I thought I would prefer the hyperlinks ... after reading the questions, it became apparent that the toolbar and voice interface would simplify the task.''
>
> ``... when you wanted to know all the states for the Republicans, then you would have to click on every single link. It would just get annoying after a while. You'd just give up I think. There'd be no way.''
>
> ``I guess I would have had to ... wow, check every state.''

demonstrated understanding of how Extempore works (e.g., input expansion):

> ``Its the easiest way cause there is only one Representative from District 17 in Florida and it takes you straight to the page.''

> ``If you click on the state then you get choices of House and whatever, but if you type in District 2 and it just goes right there.''

presented advantages and judgments:

> ``... allowed multiple pieces of information to be input at one time.''

> ``As much surfing as I do, it sort of makes me wish I had those options sometimes ya know instead of going to search engines and fooling around ... having to come up with different search criteria ....''

and also brought out frustrations:

> ``The voice interface feels a little awkward since I am not used to talking to myself ....''

> ``I don't always trust the results, [so I went back] confirming that they are all Republican.''

Many users learned that out-of-turn interaction is best suited when they have a specific goal in mind, and not meant for exploratory information-seeking (as is browsing). For instance,

> ``if I wanted to go the whole way down to a specific person, I would use [Extempore], but if I was just looking around, I would use the links.''

> ``[Extempore] is good when you know the site and know you have to go several layers deep. Links [are good] when you don't know the layout or don't know exactly what you want.''

Discussion

Extempore enables a novel approach to interact with Websites. Users with out-of-turn partial input can employ Extempore to enhance their browsing experiences. Thus, out-of-turn interaction is intended to complement browsing, and not replace it. For designers, Extempore augments their sites with capabilities for personalized interaction, without hardwiring in-turn mechanisms (as is commonly done). In addition, since usage of Extempore is optional, it preserves any existing modes of information-seeking.

There are significant lessons brought out by our study, which we only briefly mention here. This work validates our view of Web interaction as a flexible dialog and shows that users actively interleaved out-of-turn interaction with browsing. Importantly, users were proficient at determining when out-of-turn interaction is called for. Studying the rationale and usage patterns has generated a body of knowledge that can be used, among other purposes, for introducing out-of-turn interaction in new settings and to new participants. Furthermore, we have seen that it is easy to target out-of-turn interaction in domains where tasks involve

combinations of focused and exploratory behavior. Recall also that dialogs with purely declarative specifications are readily supported; others such as Task H will require further study.

Out-of-turn interaction is most effective when users have a basic understanding of the application domain and know what aspects are addressable. When users do not know what to say (Yankelovich 1996), our facility to enquire about legal utterances may induce information overload in large sites. While we have not encountered this problem in our PVS study, we suspect that applying out-of-turn interaction in large Web directories (e.g., Open Directory Project) will involve new research directions.

References

- Speech application language tags (SALT) specification. Technical report, SALT Forum, July 2002. Version 1.0.
- Allen, J.F., Guinn, C.I., & Horvitz, E. (1999). Mixed-initiative interaction. *IEEE Intelligent Systems*, **14**(5), 14-23.
- Belkin, N.J., Cool, C., Stein, A., & Thiel, U. (1995). Cases, scripts, and information seeking strategies: on the design of interactive information retrieval systems. *Expert Systems with Applications*, **9**(3), 379-395.
- Bhavnani, S.K., Bichakjian, C.K., Johnson, T.M., Little, R.J., Peck, F.A., Schwartz, J.L., & Strecher, V.J. (2003) Strategy hubs: next-generation domain portals with search procedures. In Gilbert Cockton & P. Korhonen, (Eds.), *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI)* (pp. 393-400). New York: ACM Press.
- Brusilovsky, P. (2001). Adaptive hypermedia. *User Modeling and User-Adapted Interaction*, **11**(1-2), 87-110.
- Dumais, S., Cutrell, E., & Chen, H. (2001). Optimizing search by showing results in context. *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI)*, 277-284, New York: ACM Press.
- Hearst, M.A. (2000). Next generation Web search: setting our sites. *IEEE Data Engineering Bulletin*, **23**(3), 38-48, 2000.
- Hearst, M.A., Elliott, A., English, J., Sinha, R., Swearingen, K., & Yee, K.-P. (2002). Finding the flow in Website search. *Communications of the ACM*, **45**(9), 42-49, 2002.
- Lieberman, H., Fry, C., & Weitzman, L. (2001). Exploring the Web with reconnaissance agents. *Communications of the ACM*, **44**(8), 69-75, 2001.
- Marchionini, G. (1995). *Information seeking in electronic environments*. Cambridge: Cambridge University Press.
- Olston, C., & Chi, E.H. (2003). ScentTrails: integrating browsing and searching on the Web. *ACM Transactions on Computer-Human Interaction*, **10**(3), 177-197.
- Perugini, S., & Ramakrishnan, N. (2003). Personalizing Websites with mixed-initiative interaction. *IEEE IT Professional*, **5**(2), 9-15.
- Pitkow, J., Schütze, H., Cass, T., Cooley, R., Turnbull, D., Edmonds, A., Adar, E., & Breuel, T. (2002). Personalized search. *Communications of the ACM*, **45**(9), 50-55.

- Ricca, F., & Tonella, P. (2001). Web application slicing. In *Proceedings of the IEEE International Conference on Software Maintenance (ICSM)*, 148-157, Los Alamitos: IEEE Computer Society.
- Robertson, G.G., Cameron, K., Czerwinski, M., & Robbins, D. (2002). Polyarchy visualization: visualizing multiple intersecting hierarchies. In Loren G. Terveen & D.R. Wixon, (Eds.), *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI)* (pp. 423-430). New York: ACM Press.
- Sacco, G.M. (2000). Dynamic taxonomies: a model for large information bases. *IEEE Transactions on Knowledge and Data Engineering*, **12**(3), 468-479.
- Yankelovich, N. (1996). How do users know what to say? *ACM Interactions*, **3**(6), 32-43.